

APPARATUS AND METHOD FOR STOPPING ITERATIVE DECODING
IN A CDMA MOBILE COMMUNICATION SYSTEM

PRIORITY

5

This application claims priority to an application entitled "Apparatus and Method for Stopping Iterative Decoding in a CDMA Mobile Communication System" filed in the Korean Industrial Property Office on December 23, 2000 and assigned Serial No. 2000-81387, the contents of which are hereby
10 incorporated by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

15

The present invention relates generally to an apparatus and method for stopping iterative decoding in a CDMA (Code Division Multiple Access) mobile communication system, and in particular, to an apparatus and method for stopping iterative decoding caused by an occurrence of an error.

20

2. Description of the Related Art

In general, a CDMA mobile communication system refers to a mobile communication system that performs radio communication by adopting the CDMA technique. Such a mobile communication system performs forward error correction (FEC) in order to correct errors caused by the noises generated in
25 transport channels. For the forward error correction, the CDMA mobile communication system typically uses a convolutional code or a turbo code.

In particular, the turbo code has been adopted as an error correction code in both a synchronous 3GPP2 (3rd Generation Partnership Project 2) system and
30 an asynchronous 3GPP (3rd Generation Partnership Project) system, both recently

attracting attention as a 3rd generation mobile communication system. It is well known that the turbo code is superior in performance to the existing convolutional code, especially during high-speed data transmission. In addition, the turbo code, a kind of the FEC code, is advantageous in that it can increase a reliability of transmission data by effectually correcting an error caused by the noises generated in the transport channels.

An existing convolutional encoder, however, outputs decoded data after performing decoding once on each input frame. In contrast, a turbo encoder generally uses an iterative decoding scheme for outputting decoded data after performing decoding several times on each input frame.

It is very important for the iterative decoding scheme to determine the time for stopping the decoding. In order to reduce a decoding time and minimize power consumption for the decoding, the iterative decoding scheme must frequently check an occurrence of the error for a decoding frame. For this purpose, a turbo code including a CRC (Cyclic Redundancy Check) bit is generally used.

In the CRC-based iterative decoding, a transmitter adds CRC bits to each input frame received at a turbo encoder, and a receiver then performs error detection using the CRC bits on a frame output in every iterative decoding process (or at every iteration). If no error has occurred in the frame in the error detection process, the receiver stops the decoding on the frame.

The CRC-based error detection can detect almost all errors by increasing the number of the CRC bits, thus making it possible to minimize an increase in a frame error rate (FER) due to the stop of the decoding. However, when applied to the actual system, the CRC-based error detection has several problems. A detailed description of the problems will be given below.

First, the 3GPP system reduces unnecessary iterations (or iterative decoding processes) by checking a frame error at every iteration at the receiver, and also inserts a CRC bit in each transport block in order to measure reliability of the decoded data at the receiver. However, when the CRC bit is inserted in each transport block, the transmitter concatenates transport blocks on its transport channel and segments the concatenated transport block into code blocks, so that the CRC bits may be separated from their associated transport blocks during encoding. This problem will be described in detail with reference to FIGs. 1 and 2. FIG. 1 is a block diagram illustrating a channel multiplexing process and a channel coding process used in the 3GPP downlink. FIG. 2 illustrates an exemplary method for segmenting a concatenated transport block into code blocks for encoding.

The CRC-related parts of FIGs. 1 and 2 will be described herein below. As illustrated in FIG. 2, the concatenated transport block is segmented into code blocks. As a result, the CRC bit may be separated from its associated transport block as stated above, in the code block segmentation process. Therefore, in the case of a non-CRC code block, the CRC can no longer be used as a stopping criterion in the iterative decoding process.

In FIGs. 1 and 2 where the 3GPP system uses the turbo code as an error correction code, if the size (or length) X_i of a concatenated transport block of an i^{th} transport channel is larger than 5114 bits, the concatenated transport block is segmented into $C_i = \lceil X_i / 5114 \rceil$ code blocks. Here, the size K_i of each code block is $K_i = \lceil X_i / C_i \rceil$ bits, and for the insufficient bits, 0-value are inserted in the starting portion bits of the first code block. The code blocks created in this way are transmitted over the transport channel after performing their turbo encoding process. The receiver then performs turbo decoding on the received code blocks.

The turbo decoding is performed in a unit of the code block determined by the transmitter. Therefore, as illustrated in FIG. 2, the CRC bits added to the rear of the transport block may be separated from their transport blocks in the code block segmentation process. In the case of the non-CRC code blocks, the CRC bits can no longer be used as a criterion for stopping the decoding in the turbo decoding process. Therefore, the existing 3GPP system using the CRC as a stopping criterion for the turbo decoding has the above problem.

Second, in the existing 3GPP system using the CRC as a criterion for stopping the turbo decoding, when the code block, i.e., a frame input to the turbo decoder, is small in size, the CRC bits used in each frame effect overload during data transmission, thus reducing transmission efficiency. For example, if the size of the input frame received per second is N and the total number of the CRC bits in use is C , then the actual data transmission efficiency becomes $N/(N+C)$. Therefore, if the C is not greatly smaller than N , the transmission efficiency will be reduced to far below 1.0.

Third, in the case of the existing 3GPP or 3GPP2 system using the CRC as a criterion for stopping the turbo decoding, if the turbo decoder mis-recognizes an erroneous codeword as an error-corrected codeword due to excessive errors generated in the transport channel (this state is called "undetectable error event"), then the CRC will continuously indicate error detection at every iteration (or in every iterative decoding process). Therefore, the turbo decoder continuously iterates the decoding as many times as the maximum iteration number, and even after the iterations, outputs an erroneous codeword. In this case, it is preferable to stop the decoding by detecting the "undetectable error event" before the decoding is iterated as many times as the maximum iteration number. However, it is not possible to detect the "undetectable error event" by simply using only the CRC bits. Further, in this case, the turbo decoder must iterate the decoding as many times as the maximum iteration number, thus increasing the power consumption

and the decoding time.

In order to solve the above problems, research has been made into a method of stopping iteration of the decoding on the turbo code without using the
 5 CRC. For example, the proposed method is to use the minimum absolute value of log likelihood ratio (LLR) values, which are output from the turbo decoder at every iteration in the iterative decoding process for the turbo code. That is, this method stops the decoding if the minimum value of absolute LLR values output at every iteration is higher than a threshold. This method shows relatively
 10 excellent FER performance and reduces an average decoding iteration number, thus contributing to a decrease in the power consumption. In addition, it is very easy to realize the method by hardware. However, in the above-stated method, the threshold for guaranteeing the excellent FER performance depends upon E_b/N_0 and the frame size. Accordingly, the threshold must be sensitively
 15 changed in response to a change in the values (of E_b/N_0 and the frame size). Particularly, in the actual system having difficulty in accurately measuring E_b/N_0 of a received signal, it is very difficult to accurately estimate a change in the values, caused by a change in the transmission environment, and thus, the method using the minimum value of absolute LLR values with existing threshold
 20 is also not proper to be used as stopping criteria for the iterative decoding.

The problems, which may occur in the error correction method using the turbo code, will be summarized below.

25 First, the problems, which may occur in the method for stopping the turbo decoding using the CRC, are as follows.

(1) In the 3GPP system, the CRC bits for the transport blocks can be separated from their associated transport blocks in the code block segmentation
 30 process, so that the CRC can no longer be used as a criterion for error detection.

(2) If the frame input to the turbo encoder is small in size (i.e., if the transport block is small in size in the 3GPP system), the CRC bits used in each input frame may become overload, causing a transmission rate loss.

5

(3) When the turbo decoder mis-recognizes an erroneous codeword as an error-corrected codeword due to excessive errors generated in the transport channel, the CRC will continuously indicate error detection at every iteration. Therefore, the turbo decoder continuously iterates the decoding as many times as the maximum iteration number, and even after the iterations, outputs an erroneous codeword. In this case, it is preferable to stop the decoding by detecting the “undetectable error event” before the decoding is iterated as many times as the maximum iteration number. However, it is not possible to detect the “undetectable error event” by simply using only the CRC. Further, in this case, the turbo decoder must iterate the decoding as many times as the maximum iteration number, thus increasing the power consumption and the decoding time.

Second, the problems, which may occur in the method for stopping the turbo decoding using the LLR values, are as follows.

20

(1) In the actual transport channel environment, there is a need for a decoding stopping criterion, which is independent of the frame size and can be obtained without measuring E_b/N_0 . However, in the conventional method, the stopping criterion is dependent upon the values of the frame size and E_b/N_0 .

25

(2) In order to stop the iterative decoding while obtaining meaningful FER performance, it is necessary to change a threshold used in the LLR-based turbo decoding stopping method according to the E_b/N_0 and the frame size. However, in the actual environment, it is very difficult to implement a proper estimator.

30

SUMMARY OF THE INVENTION

It is, therefore, an object of the present invention to provide an error
5 correction apparatus and method for more effectually stopping iterative decoding
in a CDMA mobile communication system.

It is another object of the present invention to provide an error correction
apparatus and method having a stopping criterion for reducing a decoding time
10 for iterative decoding by a decoder in a CDMA mobile communication system
using a turbo code.

It is further another object of the present invention to provide an error
correcting apparatus and method for reducing power consumption caused by
15 iterative decoding by a decoder in a CDMA mobile communication system using
a turbo code.

It is yet another object of the present invention to provide an error
correction apparatus and method using a log likelihood ratio (LLR) as a criterion
20 for stopping turbo decoding in a CDMA mobile communication system using a
turbo code.

It is still another object of the present invention to provide an error
correction apparatus and method capable of stopping turbo decoding even when a
25 CRC bit is separated from its transport block due to frame segmentation, in a
CDMA mobile communication system.

It is still another object of the present invention to provide an error
correction apparatus and method for updating a threshold, criterion for
30 determining a stop of decoding, using a minimum absolute value of LLR values

in a CDMA mobile communication system.

It is still another object of the present invention to provide an error correction apparatus and method using both of CRC-based stopping criterion and
 5 a LLR-based stopping criterion at every iteration in a CDMA mobile communication system.

To achieve the above objects, the present invention provides a novel stopping criterion for stopping turbo decoding to prevent unnecessary iterative
 10 decoding in a CDMA mobile communication system using a turbo code, when a frame input to a turbo encoder has no CRC bit or failed CRC bits. The novel stopping criterion can be simply realizable by hardware and equally applicable regardless of E_b/N_0 and a frame size. Further, in order to reduce a decoding time and power consumption despite the use of the CRC bits for error correction in the
 15 frame input to the turbo encoder, the present invention can implement an error correction apparatus and method selectively using both of the novel stopping criterion and CRC-based stopping criterion.

According to a first aspect of the present invention, there is provided a
 20 method for stopping iterative decoding in a turbo decoder performing iterative decoding on a received frame comprised of information bits and then outputting the iteratively decoded results. The method comprises sequentially outputting absolute LLR values associated with the respective information bits of the received frame by the iterative decoding; selecting a minimum value $M(i)$ among
 25 the sequentially output absolute LLR values in i^{th} decoding process; issuing a command to stop the iterative decoding after additionally performing the iterative decoding once, if the minimum value $M(i)$ is larger than a first threshold $T_1(i)$ determined based on a minimum value F_{\min} among absolute LLR values output through previous iterative decoding; and stopping the iterative decoding on the
 30 received frame in response to the command, and outputting the decoded results at

the stop point.

According to a second aspect of the present invention, there is provided a method for stopping iterative decoding in a turbo decoder performing iterative
 5 decoding on a received frame comprised of information bits and then outputting the iteratively decoded results. The method comprises sequentially outputting absolute LLR values associated with the respective information bits of the received frame by the iterative decoding; selecting a minimum value $M(i)$ among the sequentially output absolute LLR values in i^{th} decoding process; issuing a
 10 command to stop the iterative decoding when the selected minimum value $M(i)$ is satisfied with both of the following two conditions, (1) the one is a case where the minimum value $M(i)$ is larger than a first threshold $T1(i)$ determined based on a minimum value F_{\min} among minimum values $M(i)$ output through previous iterative decoding, and (2) the other is a case where the minimum value $M(i)$ is
 15 larger than a second threshold $T2(i)$ determined based on a minimum value I_{\min} among minimum values $M(i)$ satisfying the aforesaid condition (1); and stopping the iterative decoding in response to the command and then outputting the decoded results at the stop point.

20 According to a third aspect of the present invention, there is provided an apparatus for stopping iterative decoding in a turbo decoder performing iterative decoding on a received frame comprised of information bits and then outputting the iteratively decoded results. The apparatus comprises a turbo decoder for sequentially outputting absolute LLR values associated with the respective
 25 information bits of the received frame by the iterative decoding, and stopping the iterative decoding in response to a stop command for the iterative decoding; a minimum absolute LLR detector for selecting a minimum value $M(i)$ among the sequentially output absolute LLR values; a comparison selector for comparing said minimum value $M(i)$ with the first threshold and updating threshold in
 30 accordance with the minimum absolute LLR values output from the minimum

absolute LLR detector; and a controller for issuing a command to stop the iterative decoding, if the minimum value $M(i)$ is larger than a first threshold determined based on a minimum value F_{\min} among absolute LLR values output through previous iterative decoding.

5

According to a fourth aspect of the present invention, there is provided an apparatus for stopping iterative decoding in a turbo decoder performing iterative decoding on a received frame comprised of information bits, and then outputting the iteratively decoded results. The apparatus comprises a turbo
10 decoder for sequentially outputting absolute LLR values associated with the respective information bits of the received frame by the iterative decoding, and stopping the iterative decoding in response to a stop command for the turbo decoding; a minimum absolute LLR detector for selecting a minimum value $M(i)$ among the sequentially output absolute LLR values; a comparison selector for
15 comparing said minimum value $M(i)$ with a first threshold and in sequence comparing said minimum value $M(i)$ with a second and updating threshold in accordance with the minimum absolute LLR values output from the minimum absolute LLR detector; and a controller for issuing a command to stop the iterative decoding, if the minimum value $M(i)$ is larger than a first threshold $T_1(i)$
20 determined based on a minimum value F_{\min} among absolute LLR values output through previous iterative decoding and is also larger than a second threshold $T_2(i)$ determined based on I_{\min} which is the minimum value among minimum absolute LLR values that has satisfied the first stop condition in up to the previous decoding process for a currently decoded frame.

25

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objects, features and advantages of the present invention will become more apparent from the following detailed description
30 when taken in conjunction with the accompanying drawings in which:

FIG. 1 illustrates a process for creating a downlink channel in a general CDMA mobile communication system;

FIG. 2 illustrates an exemplary method for segmenting a concatenated transport block into code blocks in the general CDMA mobile communication
5 system;

FIG. 3 illustrates a structure of a turbo decoder to which a stopping criterion according to the present invention is applied;

FIG. 4 illustrates a detailed structure of the minimum absolute LLR detector of FIG. 3;

10 FIG. 5 illustrates a flow chart of stopping iterative decoding by a turbo decoder according to an embodiment of the present invention;

FIG. 6 illustrates a structure of a turbo decoder to which the stopping criterion according to another embodiment of the present invention is applied;

FIGs. 7 and 8 illustrates simulation results according to the different
15 embodiments of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A preferred embodiment of the present invention will be described herein
20 below with reference to the accompanying drawings. In the following description, well-known functions or constructions are not described in detail since they would obscure the invention in unnecessary detail.

The present invention has been proposed considering the fact that a
25 change in the minimum absolute value of LLR values is variable depending on the type of error generated in a received frame in the system using a turbo code. That is, when a turbo decoder iteratively decodes a frame, several types of errors shall be generated in the frame. The error types are classified into a "correctable error", a "detectable error" and an "undetectable error". Further, probabilities of
30 the errors are also classified into "correctable error probability", "detectable error

probability” and “undetectable error probability”. In addition, a change in the minimum absolute LLR value, caused by the iterative decoding, has the characteristic described below.

5 1. Correctable Error

As a received frame is subject to iterative decoding, it converges into a specific codeword. When the received frame has a correctable error, the codeword corresponds to a correctly decoded codeword. Thus, the decoding result will remain unchanged, even though the frame is further subject to the
10 iterative decoding. That is, the frame is decoded into a correct codeword transmitted from the transmitter. In this case, the minimum absolute LLR value increases with the decoding iteration number, because the reliability of the decoded codeword is represented by LLR.

15 2. Undetectable Error

Like in the case of the correctable error, as a received frame is subject to iterative decoding, it converges into a specific codeword and location and quantity of error in decoded frame is fixed with the decoding iteration number. However, the codeword is a codeword affected by an error generated in the
20 correct codeword transmitted by the transmitter, so that it is not possible to correct this error even though the received frame is further subject to the iterative decoding. This error event occurs due to excessive errors on the transport channel. In this case also, the minimum absolute LLR value increases with the decoding iteration number. Thus, this error is indistinguishable from the correctable error.

25

3. Detectable Error

Unlike in the case of the correctable error and the undetectable error, as a received frame is subject to iterative decoding, it continuously transitions between a plurality of codewords, rather than converging into a specific
30 codeword. Therefore, as the result of the decoding, a relatively large number of

errors are generated, and the number of errors is not remarkably decreased but continuously changed, even though the iterative decoding is continuously performed. Even this error event occurs due to the excessive errors on the transport channel, the minimum absolute LLR value is very small value and the
 5 minimum absolute LLR value scarcely increases in spite of an increase in the decoding iteration number. Thus, this error is distinguishable from the correctable error and the detectable error.

The turbo decoding stopping criterion (hereinafter, referred to as
 10 “stopping criterion” for short) according to the present invention divides a frame input to the turbo decoder into a type A, which corresponds to the first and second events among the above three error events, and a type B, which corresponds to the third event, based on the minimum absolute LLR value output at every iteration. Therefore, if it is considered that the input frame corresponds
 15 to the type A, the turbo decoding is stopped without iteration of the decoding any longer. Here, there are two available methods for stopping the turbo decoding, and those methods will be described herein below.

Describing the first method, if the minimum absolute LLR value of a
 20 frame is larger than a predetermined threshold in an iterative decoding process for the frame input to the turbo decoder, the frame is considered as a frame corresponding to the type A. In order to obtain stable FER performance based on the frame size and E_b/N_0 , criteria indicating a reception state of a channel, the threshold is not fixed depending on the E_b/N_0 and the frame size, but determined
 25 by calculating minimum absolute LLR values created in the previous iterative decoding process and then adding an adding factor, which will be described later, to the minimum value selected from the calculated values. If the above condition is satisfied, the decoding is iterated once again before stopping the decoding. This is to prevent the uncorrected error caused by previously stopping the
 30 decoding based on only the above stop condition, from affecting the FER

performance in a region where E_b/N_0 is high. Herein, this will be referred to as a “first stop condition”.

Describing the second method, in order to guarantee improved FER
 5 performance, if a selected minimum absolute LLR value is satisfied with the first threshold, then subsequently it is checked that the selected minimum absolute LLR value is also satisfied with the second threshold. And then, when checked that the selected minimum absolute LLR value is satisfied with both the first and second threshold, the controller stops the iterative decoding. Herein, the second
 10 threshold is determined based on I_{\min} , which is the minimum value among minimum absolute LLR values that has satisfied the first stop condition in up to the previous decoding process for a currently decoded frame. Herein, this will be referred to as a “second stop condition”. By supplementing such an algorithm, it is possible to stop the turbo decoding without a loss of the FER performance
 15 regardless of the E_b/N_0 and the frame size, in performing iterative decoding on the turbo code.

Prior to a detailed description of the present invention, a definition of the terms used herein will be given below.

20

The term “frame length (FL)” refers to the size of an input frame provided to the turbo encoder, i.e., the number of information word bits that one input frame include. “ k ” ($0 \leq k \leq FL$) indicates the sequence of the information word bits constituting the input frame, and “MAX_ITERATION” indicates the
 25 maximum decoding iteration number previously determined by the turbo decoder. “ i ” ($0 \leq i \leq \text{MAX_ITERATION}$) represents a current decoding iteration number, while “ j ” ($0 \leq j \leq i$) represents a previous decoding iteration number. Further, “LLR(k)” represents a soft decision output of the turbo decoder for a k^{th} information word bit of one frame. “MAX” is a constant determined to be larger

than all available LLR values, and is a value for initialization when the algorithm starts its operation. The constant MAX does not greatly affect the operation of the algorithm according to the present invention. “ T_f ” is an adding factor for a threshold of a stopping criterion used for the first turbo decoding stop test, while

5 “ T_d ” is an adding factor for a threshold of a stopping criterion used for the second turbo decoding stop test. “ F_{\min} ” represents the minimum value among minimum absolute LLR values calculated in up to the previous decoding process for a currently decoded frame, and “ I_{\min} ” represents the minimum value among minimum absolute LLR values that has satisfied the first stop condition in up to

10 the previous decoding process for a currently decoded frame. “ $T_1(i)$ ” indicates a threshold used for the first turbo decoding stop test, and “ $T_2(i)$ ” indicates a threshold used for the second turbo decoding stop test. Finally, “ m_flag ” ($m_flag=1$ or 2) means a factor indicating a mode of the turbo decoding stop test.

15 An outline of the overall algorithm for the new turbo decoding stopping method according to the present invention has been described, and the detailed description of each part will be given later. The new turbo decoding stopping method according to the present invention comprises a measure and thresholds, given below.

20

1. Measure of Stopping Criterion $M(i)$

This is a value measured at every iteration for an efficient decoding stop in the iterative turbo decoding process. This value is defined as a minimum absolute LLR value output at the i^{th} iteration.

25

2. First Threshold of Stopping Criterion $T_1(i)$

This is a first threshold for stopping the iterative decoding. If the $M(i)$ satisfies the first turbo decoding stop condition exceeding the first threshold $T(i)$, the decoding is stopped after one additional iteration, or it is determined whether

30 the second turbo decoding stop condition is satisfied.

3. Second Threshold of Stopping Criterion $T_2(i)$

This is a second threshold for stopping the iterative decoding. If the $M(i)$ satisfies the second threshold $T_2(i)$ as well as the first threshold $T_1(i)$, the
5 decoding is stopped.

A method for calculating the measure $M(i)$ and the thresholds $T_1(i)$ and $T_2(i)$ will be described below. The $M(i)$ and the first and second thresholds $T_1(i)$ and $T_2(i)$ are calculated from absolute LLR values $|LLR(k)|$ associated with the
10 respective information word bits at each iteration. In Equation (1) below, “min” represents a function for calculating the minimum value, and “ $0 \leq k \leq FL-1$, iteration= i ” means that k ranges from 0^{th} bit to $FL-1^{th}$ bit while calculating the minimum value at the i^{th} iteration, i.e., the minimum value is calculated for all information word bits of the input frame. That is, Equation (1) represents that an
15 absolute LLR value $|LLR(k)|$ with the minimum k ($0 \leq k \leq FL-1$) at the i^{th} iteration is selected as $M(i)$.

Equation (1)

$$M(i) = \min_{\substack{0 \leq k \leq FL-1 \\ \text{iteration}=i}} |LLR(k)|$$

$$20 \quad T_1(i) = \min_{1 \leq j \leq i-1} \left[\min_{\substack{0 \leq k \leq FL-1 \\ \text{iteration}=j}} |LLR(k)| \right] + T_f, \quad T_1(0) = MAX$$

$$T_2(i) = Imin + T_d$$

In Equation (1), $M(i)$ is the minimum absolute value among the absolute LLR values output from the frame, which is currently decoded at the i^{th} iteration.
25 $T_1(i)$ is a value determined by adding an adding factor T_f to the minimum absolute value among the absolute LLR values calculated in up to the $(i-1)^{th}$ iterative decoding process, and $T(2)$ is a value determined by adding an adding

factor T_d to the $M(i)$ value for the case where the first turbo decoding stop test is passed. The embodiment of the present invention does not use the thresholds $T_1(i)$ and $T_2(i)$, the stopping criteria, as the constants previously determined based on the E_b/N_o , so that the stopping criteria of the turbo decoder operates
 5 regardless of the E_b/N_o and the frame size. Instead, the thresholds $T_1(i)$ and $T_2(i)$ are updated using information on the LLR values continuously output during iteration of the turbo decoding. Since the LLR values represent transport channel situation therein, it is not necessary to previously determine the thresholds depending on E_b/N_o .

10

FIG. 3 illustrates a structure of a turbo decoder to which a stopping criterion according to the present invention is applied. Referring to FIG. 3, a soft input buffer 310 buffers input frames received over a transport channel in a frame unit. A decoding part 320 performs turbo decoding on the frames output from the
 15 soft input buffer 310. Further, the decoding part 320 determines whether to stop the turbo decoding based on a stopping criterion, and provides the turbo-decoded values to an output buffer 330. The decoding part 320 includes a turbo decoder 322, a minimum absolute LLR detector 324, a comparison selector 326 and a controller 328. The turbo decoder 322 performs error correction and decoding,
 20 through turbo decoding, on the frame bits output from the soft input buffer 310, and provides the decoded results to the output buffer 330 in response to a stop signal STOP_TURBO from the controller 328. In addition, the turbo decoder 322 calculates absolute LLR values $|LLR(k)|$ associated with the respective frame bits output from the soft input buffer 310. The minimum LLR detector 324
 25 receives the $|LLR(k)|$ values of the currently decoded frame, provided from turbo decoder 322, and an initial maximum value MAX and a select signal SEL, provided from the controller 328, and then outputs a minimum absolute LLR value $(\min |LLR(k)|, M(i))$, among the $|LLR(k)|$ values based on the MAX and the SEL. A detailed structure of the minimum absolute LLR detector 324 is

illustrated in FIG. 4. The comparison selector 326 compares $M(i)$ of the currently decoded frame, obtained by the minimum absolute LLR detector 324, with a threshold, and updates the F_{\min} or I_{\min} value and then registers the updated values, if the $M(i)$ is less than the threshold. The controller 328 determines whether to

5 stop the turbo decoding based on the comparison results from the comparison selector 326. Further, the controller 328 outputs the STOP_TURBO signal according to the determined results. For example, if the controller 328 determines to stop the turbo decoding, it switches the STOP_TURBO signal from '0' to '1' so that the turbo decoder 322 stops the turbo decoding and outputs the decoded

10 frame to the output buffer 330. When the turbo decoder 322 receives the STOP_TURBO signal from the controller 328, the turbo decoder 322 stop the iterative decoding and provides the decoded results to the output buffer 330. An operation of the decoding part 320 for stopping the turbo decoding is performed in accordance with a flow chart of FIG. 5.

15

As described above, the most important thing in implementing the present invention is to provide a scheme for comparing absolute LLR values $|LLR(k)|$ output in the process of decoding the code blocks. The calculated minimum $|LLR(k)|$ value is registered in a register and this value is analyzed by

20 an upper layer in a given test mode thereby to determine whether to further iterate the decoding.

FIG. 4 illustrates a detailed structure of the minimum absolute LLR detector 324 of FIG. 3. Specifically, FIG. 4 illustrates a scheme for detecting the

25 minimum value $M(i)$ among the absolute LLR values in order to stop the turbo decoding according to an embodiment of the present invention. As illustrated in FIG. 4, the scheme for calculating the minimum absolute LLR value among the absolute LLR values associated with the respective decoded frame bits is comprised of one comparator and two selectors.

Referring to FIG. 4, a first selector 410 receives an initial maximum value MAX from the controller 328 and the output $M(i)$ of the minimum absolute LLR detector 324, and selects one of the two inputs based on the select signal SEL from the controller 328. The controller 328 first provides a SEL signal for selecting the MAX, and next provides a SEL signal for selecting the output of the minimum absolute LLR detector 324. That is, the SEL signal has a value of '0' for only the first bit, and has a value of '1' for the other bits. A comparator 412 receives the output of the first selector 410 and the $|LLR(k)|$ value from the turbo decoder 322 as its inputs 'a' and 'b', respectively, compares the two inputs 'a' and 'b', and then outputs a selected signal to a second selector 414 based on the compared results. For example, if the $|LLR(k)|$ is larger than the output of the first selector 410 (i.e., $a < b$), then the comparator 412 outputs '1'. Otherwise, if the condition " $a < b$ " is not satisfied, the comparator 412 outputs '0'. The second selector 414 receives the $|LLR(k)|$ and the output of the first selector 410 at its input ports 0 and 1, respectively, and selects, as $M(i)$, one of the inputs at the input ports 0 and 1 based on the selected signal from the comparator 412. For example, if the selected signal from the comparator 412 is '0', the second selector 414 selects the input at its input port 0 as its output, and if the selected signal from the comparator 412 is '1', the second selector 414 selects the input at its input port 1 as its output. As described above, the minimum absolute LLR detector 324 of FIG. 4 initializes a value registered in the register to the maximum value MAX, and then updates the register while comparing the $|LLR|$ values associated with information word bits of the frame with the value registered in the register. As a result, the value finally remaining in the register becomes $M(i)$.

FIG. 5 illustrates a flow chart of a decoding scheme for stopping iterative decoding by a turbo decoder according to an embodiment of the present

invention. Referring to FIG. 5, if an m_flag value is set to 1, the decoding scheme according to the present invention stops the decoding after additionally performing the decoding once, provided that a first turbo decoding stop test condition is satisfied. In contrast, if the m_flag is set to 2, the decoding scheme
 5 analyzes a second turbo decoding stop test condition in the additional iterative decoding process even after the first turbo decoding stop test condition is satisfied, and then stops the decoding, provided that both of the two stop conditions are satisfied. The decoding scheme according to the present invention must satisfy the first turbo decoding stop test condition, regardless of the set
 10 m_flag value. That is, the decoding scheme cannot satisfy the second turbo decoding stop test condition unless it satisfies the first turbo decoding stop test condition.

Shown in Table 1 is a decoding stop algorithm associated with the flow
 15 chart, illustrated in FIG. 5, of the turbo decoder according to an embodiment of the present invention. In the algorithm shown in Table 1, MAX is initially set to 127. The MAX, if it is initialized to a sufficiently large value, does not affect performance of the invention.

20 Table 1

```

i = 0,  Fmin = MAX,  Imin = MAX
do
  Turbo decoding of ith iteration
  Find M(i) (minimum |LLR|)

  flag = 0  -- initialization of a flag value
  if ( M(i) > Fmin + Tf )  -- first decoding stop condition
    flag = flag + 1
  end if

```

```

Fmin = minimum M(j) for all j ≤ i

    if (flag == 1)           -- second decoding stop condition
    if ( M(i) > Imin + Td )
        flag = flag + 1
    else Imin = M(i)
    end if
end if
i = i + 1
while (flag < m_flag and i < MAX_ITERATION)
if (flag == 1 and i < MAX_ITERATION) one more turbo decoding
end if

```

In the turbo decoding stop algorithm shown in Table 1, a flag value indicating whether the minimum absolute LLR value of a frame satisfies a given turbo decoding stop condition must be larger than m_flag in order to stop the decoding. Here, the reason for using the m_flag is as follows. Of course, the m_flag value is previously determined by the user according to a decoding mode.

If the m_flag value is set to 1, it is determined whether the first turbo decoding stop test condition is satisfied. If the condition is satisfied, the turbo decoding is additionally performed once more before being stopped. However, if the m_flag value is set to 2, even the second turbo decoding stop test condition should be satisfied in the additional decoding process in order to stop the decoding. Therefore, in the embodiment of the present invention, the second turbo decoding stop test condition cannot be satisfied ($I_{\min} = \text{MAX}$) unless the first turbo decoding stop test condition is not satisfied at least once, as stated above.

An operation of the present invention will be described herein below

with reference to FIGs. 3 to 5.

Referring to FIGs. 3 to 5, in step 510 of FIG. 5, the controller 328 performs an initialization process. In the initialization process of step 510, the controller 328 determines a first adding factor T_f for a first turbo decoding stop test criterion, a second adding factor T_d for a second turbo decoding stop test criterion and an m_flag value for determining a turbo decoding stop mode. Further, in the initialization process, F_{min} and I_{min} are set to MAX, and i for counting the decoding iteration number is initialized to '0'.

10

After the initialization is completed, LLR values are calculated through turbo decoding in step 512. Specifically, the turbo decoder 322 in the decoding part 320 performs turbo decoding on the frame provided from the soft input buffer 310. Further, the turbo decoder 322 calculates LLR values associated with the respective frame bits and outputs absolute values $|LLR(k)|$ of the calculated LLR values. The $|LLR(k)|$ values are provided to the minimum absolute LLR detector 324.

After the $|LLR(k)|$ values are calculated through the turbo decoding in the step 512, the controller 328 determines in step 514 whether the turbo decoding has been iterated as many times as a predetermined maximum iteration number. The determination is carried out by comparing a value $MAX_ITERATION-1$ determined by subtracting 1 from the maximum iteration number $MAX_ITERATION$ with a current iteration number i . For example, if $i=MAX_ITERATION-1$ after the decoding is performed in the step 512, the controller 328 proceeds to step 534 where it stops the turbo decoding and then requests the turbo decoder 322 to output the results decoded in the step 512. That is, the controller 328 provides a control signal $STOP_TURBO$, i.e., a turbo decoding stop command, to the turbo decoder 322. Upon receiving the control

signal STOP_TURBO, the turbo decoder 322 outputs the turbo-decoded results to the output buffer 330.

However, if it is determined in the step 514 that the turbo decoding has not been iterated as many times as the preset number, a minimum value $M(i)$ among the $|LLR(k)|$ values calculated in the step 512 is calculated and then registered in step 516. Specifically, having been provided with the $|LLR(k)|$ values from the turbo decoder 322, the minimum LLR detector 324 calculates the $M(i)$ based on the MAX set in the initialization process and the select signal SEL provided from the controller 328. The $M(i)$ is calculated in accordance with Equation (1).

FIG. 4 illustrates a hardware structure for calculating the $M(i)$. Referring to FIG. 4, after initializing a value of the register to MAX, the minimum LLR detector 324 updates the register while comparing the $|LLR(k)|$ values associated with the information word bits of the frame with the value registered in the register, and then the value finally remaining in the register becomes $M(i)$. The calculated value $M(i)$ is provided to the comparison selector 326 and the controller 328.

20

After the $M(i)$ is calculated in step 516, the comparison selector 326 determines in step 518 whether the first turbo decoding stop test condition, $M(i) > F_{\min} + T_f$, is satisfied. That the first turbo decoding stop test condition, $M(i) > F_{\min} + T_f$, is satisfied means that the calculated value $M(i)$ is larger than a first threshold $T_1(i)$ determined by adding the T_f value set in the initialization process to the currently set F_{\min} value. The F_{\min} value is a variable, which will be described later.

If the first turbo decoding stop test condition is satisfied in the step 518,

the flow chart proceeds to step 526. Otherwise, the flow chart proceeds to step 520 to reset the F_{\min} . In the step 520, the comparison selector 326 determines whether to reset the F_{\min} by comparing the $M(i)$ with the F_{\min} . If $M(i) < F_{\min}$ in the step 520, the controller 328 sets the $M(i)$ to F_{\min} in step 522, and then
 5 proceeds to step 524. However, if the $M(i)$ is not less than the F_{\min} in the step 520, the controller 328 jumps to the step 524 without performing the process of resetting the F_{\min} . In the step 524, the controller 328 increase the i value by one, and then returns to the step 512 to iterate the turbo decoding. If the turbo decoder 322 uses a fixed point RESOVA algorithm, it fixedly sets the F_{\min} to 0, so the
 10 update process of the step 522 can be omitted. This is based on the assumption that in the case of the fixed point RESOVA algorithm, the minimum value due to iteration of the minimum absolute LLR value of a frame can approximate to 0.

In the step 526, the controller 328 determines the currently set decoding
 15 stop mode by checking the m_flag set in the initialization process. As stated above, there are two decoding stop modes. The first stop mode corresponds to the case where the m_flag is set to 1. In this case, if the first turbo decoding stop test condition is satisfied, the turbo decoding is stopped. The second stop mode corresponds to the case where the m_flag is set to 2. In this case, the second
 20 turbo decoding stop test condition as well as the first condition must be satisfied in order to stop the turbo decoding. In realizing the actual hardware, the m_flag value is not checked at every iteration, but a decoding stop mode based on an initially set value is continuously used. That is, the decoding stop mode is not changed according to the iterative decoding.

25

If the m_flag is currently set to 1, it means that the first turbo decoding stop test condition is satisfied in the step 518. Therefore, the controller 328 proceeds to step 532 where it performs additional decoding once, and then requests the stop of the turbo decoding and the output of the decoded results in
 30 the step 534. That is, if it is determined by the comparison selector 326 that the

first turbo decoding stop test condition is satisfied, after additional decoding and then the controller 328 provides the control signal STOP_TURBO, a turbo decoding stop request command, to the turbo decoder 322.

- 5 In contrast, if the m_flag is currently set to 2, the controller 328, together with the comparison selector 326, performs the processes of steps 528, 530 and 524 until the second turbo decoding stop test condition is satisfied. That is, if it is determined by the controller 328 that the m_flag is set to 2, the comparison selector 326 determines in step 528 whether the second turbo decoding stop
- 10 condition, $M(i) > I_{min} + T_d$, is satisfied. That the second turbo decoding stop condition, $M(i) > I_{min} + T_d$, is satisfied means that the calculated value $M(i)$ is larger than a second threshold $T_2(i)$ determined by adding the T_d set in the initialization process to the currently set I_{min} value. If the second turbo decoding stop test condition is satisfied in the step 528, the flow chart proceeds to step 534.
- 15 Otherwise, the flow chart proceeds to step 530 to reset the I_{min} . In step 530, the comparison selector 326 sets the $M(i)$ to the I_{min} , and then proceeds to step 524. In step 524, the controller 328 increases the i value by one, and then returns to step 512 to iterate the turbo decoding. In the case of the first iteration ($i=0$), the I_{min} is initialized to MAX, so the second turbo decoding stop test condition can
- 20 never be satisfied. Therefore, the turbo decoding should be iterated after updating the I_{min} in step 530.

However, if the second turbo decoding stop test condition is satisfied in the above operation, the controller 328 proceeds to the step 534 where it requests

25 the stop of the turbo decoding and the output of the decoded results. That is, the controller 328 provides the control signal STOP_TURBO, a turbo decoding stop request command, to the turbo decoder 322. Upon receiving the control signal STOP_TURBO, the turbo decoder 322 outputs the results decoded by the turbo decoding operation to the output buffer 330.

Next, another embodiment of the present invention will be described in detail with reference to the accompanying drawings.

Another embodiment of the present invention uses the LLR-based turbo
 5 decoding stop scheme according to the first embodiment along with the existing CRC-based turbo decoding stop scheme in order to obtain improved performance. That is, this embodiment performs both the CRC and the stop test based on the minimum absolute LLR value after decoding a frame once, thereby stopping the decoding if any one of the two stop conditions is satisfied.

10

FIG. 6 illustrates a structure of a turbo decoder to which the stopping criterion according to another embodiment of the present invention is applied. Specifically, the turbo decoder supports both the decoding stop scheme according to the first embodiment and the CRC-based decoding stop scheme.

15

Referring to FIG. 6, a turbo decoder 610 performs error correction and decoding, through turbo decoding, on an input frame received from a soft input buffer 310. A CRC checker 612 detects an error from every frame using CRC bits for the decoded results from the turbo decoder 610, each time the turbo
 20 decoding is performed by the turbo decoder 610. The CRC checker 612 outputs a result signal CRC_FLAG for the CRC check. For example, if the CRC check result is 'good', the CRC checker 612 switches the CRC_FLAG from '0' to '1' to stop the decoding. An LLR stop controller 614 has a function of testing the LLR-based decoding stop condition according to the first and second
 25 embodiments of the present invention. That is, the minimum LLR detector 324, the comparison selector 326 and the controller 328 of FIG. 3 correspond to the LLR stop controller 614. For example, if a frame satisfies the decoding stop criterion according to the present invention, the LLR stop controller 614 switches its LLR_FLAG signal from '0' to '1' to stop the decoding. A stop selection
 30 controller 616 receives the CRC_FLAG signal from the CRC checker 612 and

the LLR_FLAG signal from the LLR stop controller 614. The stop selection controller 616 selects the turbo decoding stop mode based on the two inputs. If a _FLAG (CRC_FLAG or LLR_FLAG) signal of the selected decoding stop mode is '1', the stop selection controller 616 outputs the control signal STOP_TURBO for stopping the turbo decoding being performed by the turbo decoder 610. For example, upon receiving a stop request caused by the _FLAG (CRC_FLAG or LLR_FLAG), the stop selection controller 616 switches the STOP_TURBO signal from '0' to '1' thereby to stop further decoding. The decoding stop modes are selected based on a control signal MODE_SELECT. In the case where only the CRC-based decoding stop mode is used, if the CRC_FLAG signal is changed from '0' to '1', the STOP_TURBO signal is also changed from '0' to '1', thus stopping the decoding. Further, in the case where the CRC-based decoding stop mode and the LLR-based decoding stop mode are used together, if any one of the CRC_FLAG signal and the LLR_FLAG signal is changed to '1', the STOP_TURBO signal is changed from '0' to '1', thus stopping the decoding. After stopping the decoding, the turbo decoder 610 outputs the decoded results to the output buffer 330.

Shown in Table 2 are the stop modes selected by the stop selection controller 616 in response to the MODE_SELECT signal.

Table 2

Mode	Decoding Stop Modes
00	CRC only
01	CRC or LLR (m_flag=1)
10	CRC or LLR (m_flag=2)
11	Reserved bit

As shown in Table 2, the stop selection controller 616 selects the CRC-

based decoding stop mode in response to a MODE_SELECT signal of '00', and selects the CRC-based or LLR ($m_flag=1$)-based decoding stop mode in response to a MODE_SELECT signal of '01'. Further, the stop selection controller 616 selects the CRC-based or LLR ($m_flag=2$)-based decoding stop mode in response to a MODE_SELECT signal of '10', and selects a reserved state in response to a MODE_SELECT signal of '11'. That is, another mode in addition to the above three modes can be realized by the reserved bit.

Reference will now be made to simulation results of the turbo decoder according to the preferred embodiments of the present invention.

Experiment #1

A test mode is used where both the first and second turbo decoding stop criteria are used ($m_flag=2$).

Experimental Environment #1

- AWGN Channel
- Turbo encoder: rate 1/3, 3GPP RSC encoder Recursive Systematic Convolutional encoder: component encoder for Turbo code, internal interleaver for Turbo code used in PIL interleaver 3GPP, 12 tail bits for trellis termination, interleaver size FL = 336, 1296, 4096
- Turbo Decoder: Fixed Point RESOVA Register Exchange Soft Output Viterbi Algorithm: a kind of algorithm for decoding Turbo code, MAX_ITERATION=8
- $T_f = 5$, $T_d = 10$, $m_flag = 2$
- $T_1(i) = T_f$
- CRC unused

In the iterative decoding stop algorithm used in Experiment Environment #1, a minimum value obtained by iterating minimum absolute LLR values calculated from a frame is set to 0 ($T_1(i)=T_f$, $F_{min}=0$). Thus, a process for

minimizing the F_{\min} for the second stopping criterion test can be omitted. This is based on the assumption that in the case of the fixed point RESOVA algorithm, the minimum value obtained by iteration of the minimum absolute LLR value of a frame can approximate to 0.

5

In Experiment Environment #1, a comparison is made between FER performance of the turbo code obtained based on the stopping criteria according to the invention and FER performance for which an average decoding iteration number can be obtained by Genie-aided turbo decoding. Here, “Genie-aided turbo decoding” refers to a mode for monitoring the number of errors generated in the frame by an experimenter at every iteration and stopping the decoding if no error is detected. That is, this is the most ideal case in terms of FER.

Shown in Table 3 are experimental results obtained through the above experiment.

Table 3

	Genie-aided decoding	Stopping rule
FER Performance	Best	Almost Best
Average Iteration Num	Best	Good
Actual Realizability	Impossible	Simply Realizable

FIG. 7 illustrates a comparison between FER performance of a turbo code frame obtained by the Genie-aided turbo decoding and FER performance obtained by performing decoding based on the stopping criteria according to the present invention. It is noted from FIG. 7 that the turbo decoding based on the stopping criteria according to the present invention guarantees very excellent FER performance regardless of the frame size and E_b/N_0 .

25

Shown in Tables 4 to 6 are comparisons between an average iteration number for the Genie-aided turbo decoding and an average iteration number for the turbo decoding based on the stopping criteria according to the present invention.

5

Table 4

Eb/No (dB)	Genie-aided	Stopping Rule
0.4	7.39	7.86
0.6	6.03	7.20
0.8	4.69	6.17

Table 5

Eb/No (dB)	Genie-aided	Stopping Rule
0.6	5.43	6.67
0.8	4.30	5.68
1.0	3.53	4.92

10

Table 6

Eb/No (dB)	Genie-aided	Stopping Rule
0.8	3.96	5.20
1.0	3.21	4.58
1.2	2.69	4.01
1.4	2.23	3.63
1.6	2.09	3.33

As shown in Tables 4 to 6, when the maximum iteration number is limited to 8, the turbo decoding based on the stopping criterion (rule) according to the invention is increased in its average iteration number by about 1.5 times, compared with the Genie-aided turbo decoding. The increase in the average

iteration number is a main cause of the additional one iteration, which should be necessarily performed when the measured minimum absolute LLR value satisfies the thresholds according to the present invention. However, if there is no CRC bits due to segmentation of the transport block into code blocks in the 3GPP system, it is possible to drastically reduce the power consumption and the delay time, though the decoding should be iterated as many times as the maximum iteration number.

Experiment #2

10 A test based on only the first stopping criterion ($m_flag=1$) is used along with the CRC-based test.

Experimental Environment #2

- AWGN Channel
- Turbo encoder: 3GPP2 RSC encoder (RC5), internal interleaver for Turbo code used in LCS interleaver3GPP2, 12 tail bits for trellis termination, interleaver size $FL = \{2296, 4600\}$, 16 bit CRC
- Turbo Decoder: Turbo decoding algorithm realized by applying Log operation to MAP algorithm for decoding Fixed Point LogMAPTurbo code, $MAX_ITERATION = 8$
- $T_f = 2$ ($m_flag = 1$)

15 In Experimental Environment #2, if the minimum absolute LLR value satisfies the first threshold by setting the m_flag value to 1, the decoding is stopped after one additional iteration. Further, in Experimental Environment #2, the 16-bit CRC-based stopping criteria and the stopping criteria based on the algorithm according to the present invention were simultaneously used. That is, if
 20 the CRC is 'good' in iteratively decoding a frame of the turbo code, the decoding is stopped, or if the minimum $|LLR|$ value satisfies the threshold, the decoding

is stopped after one additional iteration (CRC or LLR stop). In addition, FER and BER (Bit Error Rate) results obtained through this experiment were compared with the case where the decoding is stopped using only the CRC (CRC only stop).

5 FIG. 8 illustrates a comparison with FER and BER results of the CRC or LLR stop mode and FER and BER results of the CRC only stop mode. In FIG. 8, 'S-' represents CRC or LLR stop mode. Unlike the experiment (Experiment Environment #1) of the 3GPP system, this experiment is carried out on the assumption that there always exist the CRC bits. The experiment shows excellent
10 performance even when the turbo decoding is performed by combining the CRC only stop mode with the LLR stop mode based on the algorithm according to the present invention.

To summarize, the present invention has the following advantages.

15 First, it is possible to reduce the power consumption and the delay time by preventing unnecessary iterative decoding by providing a simple module to the turbo decoder unable to stop decoding based on the CRC bit.

20 Second, it is possible to remove overload due to the CRC bit for the turbo code with a small frame size.

25 Third, it is possible to stop decoding of the turbo decoder even without the CRC bit, if the CRC bit for detecting an error of the transport block by 3GPP turbo decoder is dispersed.

While the invention has been shown and described with reference to a certain preferred embodiment thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without
30 departing from the spirit and scope of the invention as defined by the appended

claims.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230